

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平8-263418

(43)公開日 平成8年(1996)10月11日

(51)Int.Cl.⁸

G 0 6 F 13/00

識別記号

3 5 7

庁内整理番号

7368-5E

F I

G 0 6 F 13/00

技術表示箇所

3 5 7 Z

審査請求 未請求 請求項の数5 F D (全 6 頁)

(21)出願番号

特願平7-91615

(22)出願日

平成7年(1995)3月24日

(71)出願人 000102728

エヌ・ティ・ティ・データ通信株式会社
東京都江東区豊洲三丁目3番3号

(72)発明者 高橋 宏明

東京都江東区豊洲三丁目3番3号 エヌ・
ティ・ティ・データ通信株式会社内

(72)発明者 切田 仁

東京都江東区豊洲三丁目3番3号 エヌ・
ティ・ティ・データ通信株式会社内

(72)発明者 井上 利行

東京都江東区豊洲三丁目3番3号 エヌ・
ティ・ティ・データ通信株式会社内

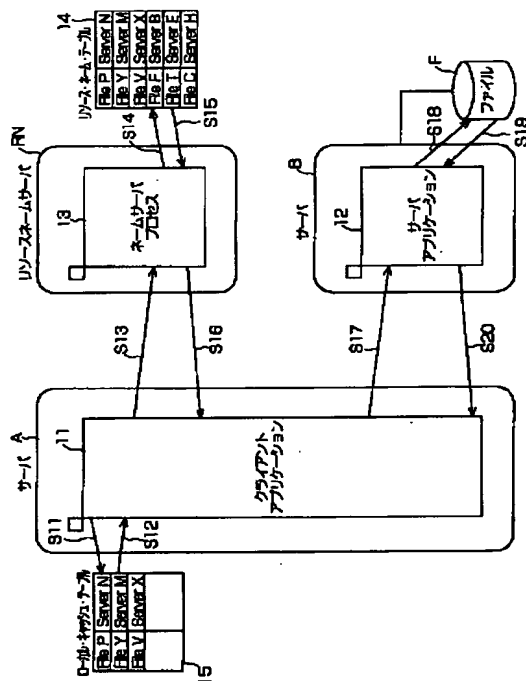
(74)代理人 弁理士 上村 輝之

(54)【発明の名称】 リソースアクセス方式及び方法

(57)【要約】

【目的】 データベース、メモリーテーブル、ファイル等のリソースを複数のサーバが分散管理する分散リソース管理システムにおいて、クライアントが所望のリソースにアクセスする場合に、クライアントがリソースを管理するサーバの名前を全く意識することなく任意のリソースにアクセスすることができるようにする。

【構成】 リソースネームサーバRNが、システム内の全てのリソースとそのリソースを管理するサーバ名とをリソースネームテーブル14上で集中管理している。クライアント11が或るリソースFにアクセスする場合、まず、そのリソースFを管理するサーバ名について、リソースネームサーバRNに問合せ(S13)、そのサーバ名「サーバB」を取得する(S16)。次に、クライアント11はその取得したサーバ名をもつサーバBに対して、リソースFへのアクセス命令を発行する(S17)。サーバ名の取得を高速に行うため、望ましくは、クライアント11はローカルキャッシュテーブル15を有し、ここに過去に問合せたリソース名とサーバ名とを登録しておく。



【特許請求の範囲】

【請求項1】 複数のサーバに分散管理されたリソースにクライアントがアクセスするためのリソースアクセス方式において、分散された個々のリソースを管理しているサーバの名前を管理するリソースネーム管理手段を備え、前記クライアントが、アクセスしようとするリソースを管理しているサーバの名前を、前記リソースネーム管理手段より取得するサーバネーム取得手段と、前記取得したサーバ名により特定されるサーバに対して、前記アクセスしたいリソースへのアクセス命令を発行するアクセス命令手段とを有することを特徴とするリソースアクセス方式。

【請求項2】 請求項1記載の方式において、前記サーバネーム管理手段が、前記リソースを管理するサーバ名の変動の事実が発生した場合、この事実の発生にตอบสนองして前記管理しているサーバ名を更新することを特徴とするリソースアクセス方式。

【請求項3】 請求項1記載の方式において、前記サーバネーム取得手段が、前記リソースネーム管理手段より既に取得したサーバ名が登録されているキャッシュテーブルを有し、前記アクセスしようとするリソースを管理しているサーバ名を取得しようとする際、先ず前記キャッシュテーブルから前記サーバ名を検索し、検索が失敗した場合に次に、前記リソースネーム管理手段より前記サーバ名を取得して前記キャッシュテーブルに登録し、前記アクセス命令手段が、前記キャッシュテーブルからの検索が成功した場合は検索されたサーバ名に基づいて、また、前記検索が失敗した場合には前記リソースネーム管理手段より取得したサーバ名に基づいて前記アクセス命令を発行することを特徴とするリソースアクセス方式。

【請求項4】 請求項3記載の方式において、前記キャッシュテーブルから検索されたサーバ名に基づいて行ったアクセスが失敗した場合、前記サーバネーム取得手段が、前記アクセスしようとするリソースを管理しているサーバ名を前記サーバネーム管理手段から取得し、前記アクセス命令手段が、前記リソースネーム管理手段より取得したサーバ名に基づいて前記アクセス命令を再度発行することを特徴とするリソースアクセス方式。

【請求項5】 複数のサーバがリソースを分散して管理するシステムにおいて、分散管理されたリソースにクライアントがアクセスするためのリソースアクセス方法において、分散された個々のリソースを管理しているサーバの名前を所定のリソースネームサーバにて管理する過程と、前記クライアントがアクセスしようとするリソースを管理しているサーバの名前を、前記リソースネームサーバ

から前記クライアントへ提供する過程と、前記クライアントより、前記提供されたサーバ名により特定されるサーバに対して、前記アクセスしたいリソースへのアクセス命令を発行する過程とを備えることを特徴とするリソースアクセス方法。

【発明の詳細な説明】**【0001】**

【産業上の利用分野】 本発明は、データベース、メモリテーブル、ファイル等のリソースを複数のサーバが分散して管理する分散リソース管理システムにおいて、クライアントが所望のリソースにアクセスするためのリソースアクセス方式の改良に関する。

【0002】

【従来の技術】 図3は従来のリソースアクセス方式の一例を示す。この例では、サーバBがリソースとしてのファイルFを管理し、サーバAがファイルFへのアクセス要求を発生するクライアントとして機能している。

【0003】 サーバAのクライアントアプリケーション1には、ファイルFを管理するサーバBの名前情報が予め組込まれている。サーバAがファイルFにアクセスする場合、クライアントアプリケーション1がサーバBの名前を指定してファイルFに対するアクセス命令を発行する（ステップS1）。すると、名前を指定されたサーバBのサーバアプリケーション2が、そのアクセス命令にตอบสนองして、ファイルFにアクセスし（ステップS2、S3）、そのアクセス結果をクライアントアプリケーション1に返信する（ステップS4）。

【0004】

【発明が解決しようとする課題】 このように従来のリソースアクセス方式では、アクセスしたいリソースを管理するサーバの名前を指定することにより、そのリソースにアクセスできるようになっている。そのため、特に分散リソース管理システムでは、リソースを管理する複数のサーバ全ての名前を組込んでクライアントアプリケーションを作成しなければならない。その結果、クライアントアプリケーションの作成作業が面倒であり生産性が悪いという問題点がある。

【0005】 また、リソースの位置が他のサーバに動的に変更されたり、新たなリソースが追加された場合には、システムを停止してクライアントアプリケーションのサーバ名の設定を変更し、再起動しなければならない。

【0006】 本発明は上記従来の問題点に鑑み、分散リソース管理方式において、クライアントがリソースを管理するサーバの名前を意識することなく所望のリソースにアクセスすることができ、その結果、クライアントアプリケーションの生産性を向上することができるリソースアクセス方式を提供することを目的とする。

【0007】

【課題を解決するための手段】 本発明に従うリソース分

散管理システムのためのリソースアクセス方式は、リソースを管理しているサーバの名前を管理するリソースネーム管理手段をシステム内に備える。そして、クライアントは、アクセスしようとするリソースを管理しているサーバの名前を上記リソースネーム管理手段より取得するサーバネーム取得手段と、この取得したサーバ名により特定されるサーバに対して、リソースアクセス命令を発行するアクセス命令手段とを有する。

【0008】また、本発明に従うリソース分散管理システムのためのリソースアクセス方法は、リソースを管理しているサーバの名前を所定のリソースネームサーバにて管理する過程と、クライアントがアクセスしようとするリソースを管理しているサーバの名前を、リソースネームサーバからクライアントへ提供する過程と、クライアントより、上記提供されたサーバ名により特定されるサーバに対して、アクセスしたいリソースへのアクセス命令を発行する過程とを備える。

【0009】

【作用】本発明によれば、システム内のどのリソースがどのサーバによって管理されているかという情報がサーバネーム管理手段によって管理されている。クライアントは、任意のリソースにアクセスしようとする場合、アクセスしたいリソースを管理しているサーバ名について、サーバネーム管理手段に問合せてこれを取得し、そのサーバ名に基づいてアクセス命令を発行する。従って、クライアントは自身でサーバ名を持たなくても、任意のリソースにアクセスできる。

【0010】また、リソースを管理するサーバに変更、追加、削除等の変動が生じた場合、サーバネーム管理手段の管理内容のみ更新するだけで簡単に対応できる。この場合、サーバネーム管理手段が、サーバの変動にตอบสนองして管理内容を更新するように構成されることが望ましい。

【0011】好適な実施例では、個々のクライアントが、リソースネーム管理手段より過去に取得したサーバ名が登録されているキャッシュテーブルを有している。そして、クライアントは、アクセスに先立ち、先ずキャッシュテーブルからアクセス対象のサーバ名を検索し、検索が失敗した場合に次に、リソースネーム管理手段よりそのサーバ名を取得する。このキャッシュテーブルの利用により、サーバ名の取得が高速に行える。

【0012】また、好適な実施例では、クライアントは、キャッシュテーブルから検索されたサーバ名に基づいて行ったアクセスが失敗した場合には、サーバネーム管理手段に問合せてアクセス対象のサーバ名を取得し、再度アクセスを実行する。これにより、リソースを管理するサーバの変動が生じた場合でも、自動的に変動後のサーバ名を指定してアクセスすることが可能になる。

【0013】

【実施例】以下、図面を参照して本発明の実施例を説明

する。図1は本発明に係る分散リソース管理システムにおけるリソースアクセス方式の一実施例を示すブロック図である。

【0014】図1において、リソースに対するアクセス要求を発生するクライアントアプリケーション11を有するサーバAと、リソースたるファイルFを管理するサーバアプリケーション12を有するサーバBとに加え、リソースを管理するサーバの名前を管理するためのリソースネームサーバRNとが設けられる。これらのサーバA、B、RNはネットワークによって相互通信可能に接続されている。

【0015】尚、図示してはいないが、本システム内には、サーバBと同様にファイルを管理しているサーバが他にも1台以上存在し、それら複数のサーバによってシステム全体のリソースが複数ファイルに分割され分散管理されている。また、サーバAと同様にリソースへのアクセス要求を発生するサーバが他に存在していても構わない。更に、全てのサーバが、サーバA及びサーバBの機能を併有していても構わない。リソースネームサーバRNの機能は、複数台のサーバによって分担することが可能ではあるが、望ましくは、システム内で1台のサーバだけがこの機能を受け持つべきである。

【0016】リソースネームサーバRNはリソースネームテーブル14を有し、このリソースネームテーブル14には、システム内の全てのファイルの名前とそれらファイルを管理するサーバの名前とが対応付けられて格納されている。

【0017】リソースネームサーバRNは、ネームサーバプロセス13によって、リソースネームテーブル14を管理する。即ち、ネームサーバプロセス13は、ファイルを管理するサーバの変更や追加や削除（以下、サーバ名の変動という）があった場合、その都度に、所定のサーバやセンタ（図示せず）からの依頼を受けて、リソースネームテーブル14の登録内容をサーバ名の変動に応じた内容に書き換える。また、ネームサーバプロセス13は、任意のサーバから特定のファイルを管理するサーバ名についての問い合わせが入力された場合、リソースネームテーブル14を参照してそのサーバ名を調べ、問合せ元のサーバに回答する。

【0018】サーバAのクライアントアプリケーション11はローカルキャッシュテーブル15を有し、リソースネームサーバRNに対し特定のファイルのサーバ名を問合せてその回答を受ける度に、そのファイル名とサーバ名とを対応付けてローカルキャッシュテーブル15に登録する。従って、ローカルキャッシュテーブル15には、過去に当該クライアントアプリケーション11がリソースネームサーバRNに問い合わせたファイル名とサーバ名とのセットが多数、その容量の許す限度内で登録されている。一方、クライアントアプリケーション11自体には、リソースを管理するサーバ名の情報は組込ま

れていない。

【0019】そして、クライアントアプリケーション11は、任意のファイルにアクセスする場合、まずローカルキャッシュテーブル15を参照することにより、また、もしローカルキャッシュテーブル15では用が足りなければ、次にリソースネームサーバRNに問合せることにより、アクセスしたいファイルを管理しているサーバ名の情報を取得し、そのサーバ名を指定してアクセス要求を発行するように構成されている。

【0020】このような構成において、例えばサーバBが管理するファイルFに対し、サーバAのクライアントアプリケーション11がアクセスしたい場合、クライアントアプリケーション11はまず、自己の管理するローカルキャッシュテーブル15を参照して、アクセスしたいファイル名「ファイルF」を検索し（ステップS11）、「ファイルF」が有れば対応するサーバ名「サーバB」をローカルキャッシュテーブル15から読出す（ステップS12）。一方、「ファイルF」がローカルキャッシュテーブル15に無ければ、クライアントアプリケーション11は次に、リソースネームサーバRNに対して「ファイルF」を管理するサーバ名を問い合わせる（ステップS13）。

【0021】リソースネームサーバRNのネームサーバプロセス13は、クライアントアプリケーション11から上記問合せを受けると、リソースネームテーブル14を参照して、問い合わせが特定したファイル名「ファイルF」に対応するサーバ名「サーバB」を取得し（ステップS4～S5）、このサーバ名「サーバB」をサーバAのクライアントアプリケーション11に返送する（ステップS16）。クライアントアプリケーション11はリ

ソースネームサーバRNからサーバ名「サーバB」を受信すると、ファイル名「ファイルF」とサーバ名「サーバB」とをローカルキャッシュテーブル15に登録する。

【0022】以上のようにして、クライアントアプリケーション11はアクセス対象を管理するサーバ名「サーバB」を、ローカルキャッシュテーブル15又はリソースネームサーバRNから取得する。すると、クライアントアプリケーション11は次に、サーバ名「サーバB」を指定してネットワークにファイルFに対するアクセス命令を発行する（S17）。

【0023】アクセス命令により名指しされたサーバBでは、サーバアプリケーション12がそのアクセス命令を受信して、指定されたファイルFにアクセスし（ステップS18、S19）、そのアクセス結果（例えば、アクセスしたレコードのアクセス後の内容）をクライアントアプリケーション11に返信する（ステップS20）。

【0024】以上のようにして、クライアントアプリケーション11は、自身ではファイルを管理するサーバ名

を一切保持してなくても、リソースネームサーバRNに問合せることにより、又は過去の間合せ結果を登録したキャッシュテーブル15を参照することにより、任意のファイルを管理するサーバ名を獲得してそのファイルにアクセスすることが可能となる。

【0025】また、サーバ名の変動があった場合にも、リソースネームサーバRNが実質的に実時間でリソースネームテーブル14を更新するため、クライアントアプリケーション11はサーバ名の変動について一切関知する必要がない。

【0026】図2は、サーバ名の変動があった場合のアクセス処理の流れを示している。以下、ファイルFの管理場所がサーバBからサーバCに移転した後において、クライアントアプリケーション11がファイルFにアクセスする場合を例にとり説明する。

【0027】既に述べたように、クライアントアプリケーション11は最初に、自己のローカルキャッシュテーブル15から「ファイルF」を検索する。もし、ローカルキャッシュテーブル15内に「ファイルF」があったとすると、そのサーバ名は従前の「サーバB」になっている。この場合、クライアントアプリケーション11は「サーバB」を指定してファイルFのアクセス要求を発行するが、するとサーバBから「アクセス失敗」の返答が返るから、次にクライアントアプリケーション11はリソースネームサーバRNに「ファイルF」のサーバ名を問合せ（ステップS21）。また、ローカルキャッシュテーブル15内に「ファイルF」が無かった場合も、同様に、クライアントアプリケーション11はリソースネームサーバRNに「ファイルF」のサーバ名を問合せ（ステップS21）。

【0028】すると、リソースネームサーバRNは、ファイルFを管理するサーバ名をリソースネームテーブル14より取得する（ステップS25）。この場合、リソースネームテーブル14は既に更新されているため、ファイルFを管理するサーバ名として「サーバC」が取得される。リソースネームサーバRNは、このサーバ名「サーバC」をサーバAに返送する（ステップS26）。

【0029】サーバAは、このサーバ名「サーバC」を受信し（ステップS22）、このサーバ名「サーバC」を指定してファイルFへのアクセス命令を発行する（ステップS23）。

【0030】サーバCは、このアクセス命令を受信して（ステップS27）、ファイルFにアクセスし（ステップS28）、ファイルFへのアクセス結果をサーバAに返送する（ステップS29）。そして、サーバAはそのアクセス結果を受信して（ステップS24）、ファイルFへのアクセス処理を終える。

【0031】以上のようにして、サーバ名の変動があっても、クライアントアプリケーション11の内容は何ら

10

20

30

40

50

変更する必要はない。ネームサーバプロセス 13 とサーバアプリケーション 12 についても、その内容を何ら変更する必要はない。

【0032】

【発明の効果】以上説明したように本発明によれば、分散リソース管理システムにおいて、クライアントがリソースを管理するサーバの名前を全く意識しなくても、任意のリソースにアクセスすることができる。

【図面の簡単な説明】

【図 1】本発明に係るリソースアクセス方式の一実施例を示すブロック図である。

【図 2】同実施例においてリソースが別のサーバに移動

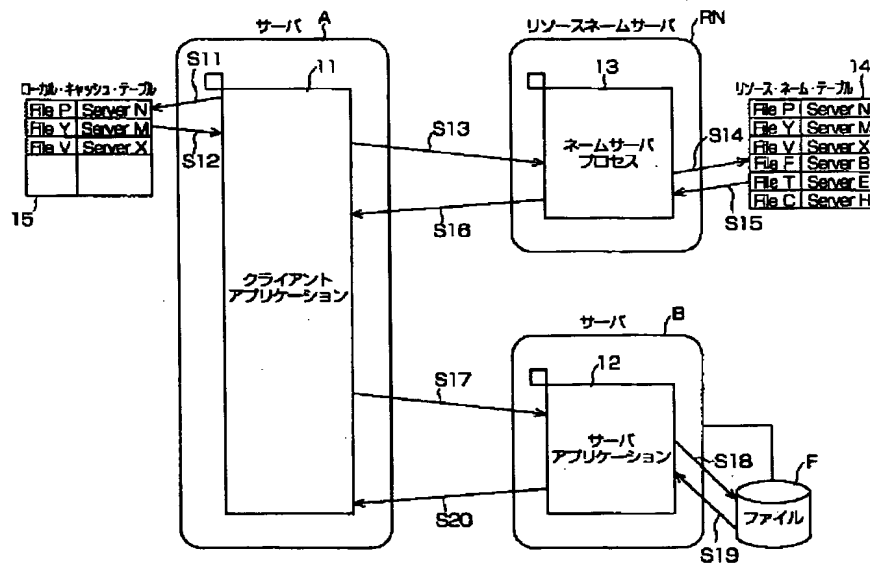
した場合のアクセス処理を示すフローチャートである。

【図 3】従来のリソースアクセス方式を示すブロック図である。

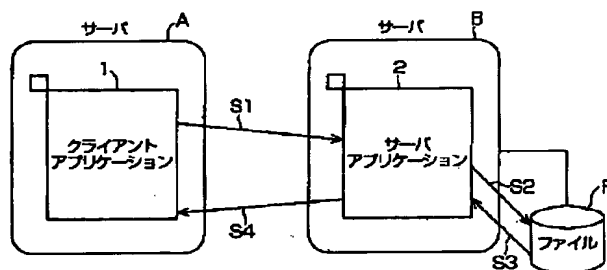
【符号の説明】

- A、B、C サーバ
- F ファイル（リソース）
- RN リソースネームサーバ
- 11 クライアントアプリケーション
- 12 サーバアプリケーション
- 13 ネームサーバプロセス
- 14 リソースネームテーブル
- 15 ローカルキャッシュテーブル

【図 1】



【図 3】



【図2】

